

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE UM CONTROLE MULTIVARIÁVEL PARA
VEÍCULO REMOTAMENTE OPERADO**

Lucas Evangelista Alves Peixoto

Gabriel Teperino Cruz

Brasília, dezembro de 2016



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE UM CONTROLE MULTIVARIÁVEL PARA
VEÍCULO REMOTAMENTE OPERADO**

Lucas Evangelista Alves Peixoto
Gabriel Teperino Cruz

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Eugenio Libório Feitosa Fortaleza, _____
ENM/UnB
Orientador

Prof. Guilherme Caribé de Carvalho, _____
ENM/UnB
Examinador interno

Prof. Carlos Humberto Llanos Quintero, _____
ENM/UnB
Examinador interno

MS.c José Oniram de A. Limaverde Filho _____
Examinador Externo

Brasília, dezembro de 2016

FICHA CATALOGRÁFICA

CRUZ, GABRIEL TEPERINO e PEIXOTO, LUCAS EVANGELISTA ALVES Implementação de um Controle Multivariável para um Veículo Remotamento Operado, [Distrito Federal] 2016. x, 41p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2016). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.	
1. Veículo Operado Remotamente	2. Controle de profundidade
3. Identificação	
I. Mecatrônica/FT/UnB	II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

CRUZ, G. T., PEIXOTO, L. E. A., (2016). Implementação de um Controle Multivariável para um Veículo Remotamento Operado. Publicação FT.TG-*n*º040/2016, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 41p.

CESSÃO DE DIREITOS

AUTOR: Gabriel Teperino Cruz e Lucas Evangelista Alves Peixoto

TÍTULO DO TRABALHO DE GRADUAÇÃO: Implementação de um Controle Multivariável para um Veículo Remotamento Operado.

GRAU: Engenheiro

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Gabriel Teperino Cruz

Lucas Evangelista Alvez Peixoto

Dedicatórias

A minha família e ao futuro

Gabriel Teperino Cruz

À minha família.

Lucas Evangelista Alves Peixoto

Agradecimentos

Agradeço a minha mãe Erondina, meu pai Wanderley, meus irmãos Luciana, Leonardo, Gabriela e Douglas, minha família que sempre me incentivou na busca dos meus sonhos e me acalentou nos momentos difíceis. Aos amigos de longa data, Marcelo “Bat” e André “Chaico” que sempre estiveram comigo e me apoiaram nessa grande jornada. Aos grandes momentos vividos com os amigos de UnB que guardarei para toda minha vida. Ao meu parceiro de trabalho final e grande amigo Gabriel, por dividir essa experiência e pelo companheirismo. E por fim, ao nosso orientador Eugênio, pela paciência, pela ajuda, pelo conhecimento e por nos confiar um projeto ambicioso.

Lucas Evangelista Alves Peixoto

Agradeço primeiramente a minha mãe Adriana, meu pai Jairton e meus irmãos Matheus e Isabela, minha família que sempre estiveram ao meu lado quando precisei, tanto nos momentos bons quanto nos ruins. Agradeço a meus avós pela infância passada com tanta alegria e pela sabedoria transmitida.

Agradeço a minha namorada Jéssica que me guiou em um novo rumo, sempre me incentivando a seguir meus sonhos. Obrigado pela paciência, pelo carinho e pelos melhores anos da minha vida.

Agradeço a meus amigos pelos momentos divertidos passados durante o período na universidade e por me proporcionarem lembranças que levarei para o resto da vida, em especial meu amigo Pedro, que emprestou sua casa e piscina e teve influência direta no desenvolvimento deste trabalho.

Por fim, agradeço ao meu parceiro de trabalho final e grande amigo Lucas, por dividir esse trabalho e pelo companheirismo, e ao nosso orientador Eugênio, pela paciência, pela ajuda, pelo conhecimento e por nos confiar um projeto ambicioso.

Gabriel Teperino Cruz

RESUMO

O veículo submarino operado remotamente ou ROV (*Remotely Operated Vehicle*) é um veículo submersível muito utilizado para realizar e supervisionar a montagem de equipamentos de exploração de petróleo e também auxiliar na manutenção dos mesmos em grandes profundidades. Neste trabalho estão detalhados os modelos matemáticos que definem a dinâmica do veículo e também todo o processo de identificação desses modelos, assim como a implementação de um sistema de controle multivariável, com o objetivo de tornar o veículo autônomo.

Palavras Chave: ROV, identificação, controle, autônomo.

ABSTRACT

The remotely operated submarine vehicle (ROV) is a submersible vehicle widely used to assemble underwater oil exploration equipments and also help with their maintenance at great depths. In this work, it is detailed the mathematical models that define the dynamics of the vehicle and also the whole process of identification of these models, as well as the implementation of a multivariable control system in order to turn it into an autonomous vehicle.

Keywords: ROV, identification, control, autonomous.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO.....	2
1.4	RESULTADOS OBTIDOS	2
1.5	APRESENTAÇÃO DO MANUSCRITO	3
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	CINEMÁTICA	4
2.2	DINÂMICA	7
2.2.1	ESFORÇOS HIDRODINÂMICOS.....	8
2.2.2	ESFORÇOS AMBIENTAIS	9
2.2.3	ESFORÇOS RESTAURATIVOS	10
2.2.4	EQUAÇÃO DINÂMICA	11
2.3	MÉTODO DOS MÍNIMOS QUADRADOS	11
2.4	PROCESSAMENTO DA IMAGEM	13
2.4.1	CONTEXTUALIZAÇÃO DO PROBLEMA	13
2.4.2	MORFOLOGIA MATEMÁTICA	14
3	IMPLEMENTAÇÃO COMPUTACIONAL.....	17
3.1	FORM1.....	18
3.2	BUTTON1_CLICK E LUZES_SCROLL	18
3.3	BUTTON1_CLICK1	18
3.4	TRACKBAR1_SCROLL.....	18
3.5	BUTTON1_CLICK_2	18
3.6	VIDEO_CLICK.....	19
3.7	ANOTADADOS	19
3.8	GRAYSCALEFILTER	19
3.9	CONTPIX	19
3.10	PIX_To_CM	20
3.11	CONTROLEPROFUNDIDADE.....	20
3.12	CONTROLEDESLOC	20
3.13	CONTROLEDISTA	20

4	DESENVOLVIMENTO DO SISTEMA DE CONTROLE	21
4.1	CONTROLE PID	23
4.2	IMPLEMENTAÇÃO	24
4.2.1	CÁLCULO DOS GANHOS	25
4.3	PROBLEMAS ENCONTRADOS	25
4.3.1	SATURAÇÃO	25
4.3.2	ALTA VARIAÇÃO DO COMANDO	26
4.3.3	PROBLEMA COM O GANHO DERIVATIVO	27
5	TESTES E RESULTADOS	28
5.1	TESTES INICIAIS PARA CONHECIMENTO BÁSICO DO ROV	28
5.2	PONTO DE LINEARIZAÇÃO E TESTE DE IDENTIFICAÇÃO	29
5.3	RESULTADOS	29
5.3.1	PRIMEIRO CONTROLE DE PROFUNDIDADE	29
5.3.2	PRIMEIRO CONTROLE MULTIVARIÁVEL	30
5.3.3	CONTROLE DE PROFUNDIDADE COM MELHORIAS	31
5.3.4	IMPLEMENTAÇÃO FINAL DO CONTROLE MULTIVARIÁVEL	33
6	CONCLUSÕES	36
	REFERÊNCIAS BIBLIOGRÁFICAS	38
	ANEXOS	39
I	DESCRIÇÃO DO CONTEÚDO DO CD	40
II	PROGRAMAS UTILIZADOS	41

LISTA DE FIGURAS

1.1	VideoRay Pro 4.	1
2.1	Sistemas de Coordenadas.....	4
2.2	Binarização da linha da imagens.....	14
2.3	Exemplo de Dilatação de Imagem.....	15
2.4	Exemplo de Erosão de Imagem	16
3.1	Arquivos Form.cs e Form.Designer.cs no Visual Studio.....	17
4.1	Evolução Temporal da Profundidade.	21
4.2	Evolução Temporal da propulsão.	22
4.3	Gráfico Evolução Temporal da Aceleração.	22
4.4	Evolução Temporal da velocidade.	23
5.1	Gráfico Propulsão Vertical x Tempo.	30
5.2	Gráfico Profundidade controlada x Tempo.....	30
5.3	Gráfico Erro x Tempo.	31
5.4	Resposta controle PI.	32
5.5	Saída do controlador PI.	32
5.6	Resposta do controlador PID com filtro de erro.	33
5.7	Comparação entre as imagens binarizadas	34
5.8	Resposta do controlador PID integrado.....	34

LISTA DE TABELAS

2.1	Graus de Liberdade e suas Variáveis.....	5
-----	--	---

LISTA DE SÍMBOLOS

Símbolos Latinos

P	Força Peso
Z_P	Profundidade
x	Componente do vetor de posição na direção <i>Surge</i>
y	Componente do vetor de posição na direção <i>Sway</i>
z	Componente do vetor de posição na direção <i>Heave</i>
u	Componente do vetor de velocidade linear na direção <i>Surge</i>
v	Componente do vetor de velocidade linear na direção <i>Sway</i>
w	Componente do vetor de velocidade linear na direção <i>Heave</i>
p	Componente do vetor de velocidade angular na direção <i>Roll</i>
q	Componente do vetor de velocidade angular na direção <i>Pitch</i>
r	Componente do vetor de velocidade angular na direção <i>Yaw</i>
X	Força na direção <i>Surge</i>
Y	Força na direção <i>Sway</i>
Z	Força na direção <i>Heave</i>
K	Força na direção <i>Roll</i>
M	Força na direção <i>Pitch</i>
N	Força na direção <i>Yaw</i>
J	Transformação não linear entre os sistema de coordenadas
m	Massa
r_G	Vetor distância entre o centro de massa do ROV e a origem do referencial móvel
I_o	Matriz que reúne momentos e produtos de inércia de um corpo
O_{xyz}	Origem do referencial móvel
M_{CR}	Matriz com as forças que envolvem o centro de massa
C_{CR}	Termo que envolve a força de Coriolis
M_A	Matriz das derivadas hidrodinâmicas
C_A	Matriz centrípeta de Coriolis
F_D	Força de arrasto
C_d	Matriz de coeficientes de arrasto hidrodinâmico
S	Área do veículo em contato com o fluido
C_{dl}	Coefficiente de arrasto linear
C_{dnl}	Coefficiente de arrasto não linear

M_C	Componente inercial dos esforços ambientais
C_C	Componente centrípeta dos esforços ambientais
g	Aceleração da gravidade
E	Empuxo hidrostático
$G()$	Componente geral para forças e momentos restaurativos
X	Profundidade, em metros
U	Constante que multiplica o termo de controle
C_1	Termo de Controle
C_2	Constante que simboliza o Peso + Empuxo
Z^2	Conjunto bidimensional de números inteiros
A, B	Um subconjunto de Z^2
$a_{n,n}$	Um elemento de A ou B
x_i, y_i, z_i	Sistema de coordenadas cartesianas para análise de imagens
N_{pb}	Número de pixels brancos
N_{pp}	Número de pixels pretos
$F3, F4, F6$	Comandos passados ao propulsor
t	Tempo
P_n	Peso atribuído para o cálculo das derivadas
U', U_0	Ponto de operação do termo de controle
a	Pólo real designado para o sistema de controle
K_P	Ganho Proporcional
K_I	Ganho Integral
K_D	Ganho Derivativo
U_0	Referência passada ao controlador

Símbolos Gregos

η_1	Vetor de posição do veículo submarino
η_2	Vetor de orientação do veículo submarino
φ	Componente do vetor de orientação na direção Roll
θ	Componente do vetor de orientação na direção Pitch
ψ	Componente do vetor de orientação na direção Yaw
ν_1	Vetor de velocidade linear
ν_2	Vetor de velocidade angular
τ_1	Vetor de forças
τ_2	Vetor de momentos
ρ	Massa específica do fluido
τ_C	Componente que compreende os esforços comparativos
α	Constante utilizada na identificação do controle
Δ	Símbolo de variação

Grupos Adimensionais

i, n Contadores

Siglas

ROV	<i>Remotely Operated Vehicle</i>
AUV	<i>Autonomous Underwater Vehicle</i>
PID	<i>Proporcional Integral Derivativo</i>
PI	<i>Proporcional Integral</i>

Capítulo 1

Introdução

1.1 Contextualização

Atualmente, mesmo com a constante busca por novas fontes de energia, o petróleo continua a ser a maior matriz energética da sociedade. Logo se torna importante buscar refinar o processo de exploração do petróleo. Um dos vários processos é a extração, que necessita de mais pesquisas, visto que o homem cada vez mais busca extrair em águas mais profundas e em solos mais difíceis de serem perfurados.

Depois de ter seu uso consolidado pela Marinha dos Estados Unidos na década de 60 viu-se no ROV (*Remotely Operated Vehicle*), um veículo submarino que possui uma câmera e pode ser controlado remotamente, uma nova alternativa para monitorar todo o processo de extração e perfuração em águas profundas de difícil acesso para o ser humano. Um ROV fabricado pela VideoRay, visto na Figura 1.1, é o objeto de pesquisa deste trabalho.



Figura 1.1: VideoRay Pro 4.

1.2 Definição do problema

Para a utilização do ROV, principalmente em ambientes adversos como o presente na exploração de petróleo, a realização de diversos treinamentos faz-se necessário, além do aprimoramento da destreza e precisão do operador. Diante de tamanhos requisitos, uma solução natural para tornar o processo mais simples e até menos custoso seria a de transformar o ROV em um AUV (*Autonomous Underwater Vehicle*). Para tal, é necessário o desenvolvimento de controladores autônomos que atuem no ROV para que o próprio execute uma tarefa pré-definida, como por exemplo, requerer que o ROV se mantenha em uma profundidade escolhida pelo operador.

Diante da complexidade e não linearidade que estão presentes no modelo dinâmico do ROV e também no processo de projeto de um controlador de profundidade para tal modelo. Controle PID (Proporcional Integral Derivativo) se apresenta como uma solução plausível e de baixa complexidade por ser bastante utilizado e disseminado não somente no meio acadêmico como na indústria.

1.3 Objetivos do projeto

Para que o ROV se comporte como veículo autônomo, é necessário aplicar o controle nos eixos em que o veículo se locomove. Este, em específico, move-se nas direções de avanço (*do inglês, surge*), guinada (*yaw*) e afundamento (*heave*). Os detalhes sobre essas direções e referências serão evidenciados posteriormente na Seção 2.

Em trabalhos anteriores foram desenvolvidos sistemas de controle PID, independentes entre si, para as direções de avanço e guinada, através do processamento de imagens da câmera do ROV [1]. Assim, o objetivo principal desse trabalho é implementar um controle PID para a direção de afundamento e assim, posteriormente, conseguir desenvolver uma estratégia para que todos os sistemas de controle atuem ao mesmo tempo.

1.4 Resultados obtidos

Os resultados presentes neste trabalho consistem basicamente na obtenção das constantes necessárias para a identificação do sistema, na realização de um controle de profundidade, e nas conclusões obtidas na tentativa de integrá-lo com os desenvolvidos por [1].

1.5 Apresentação do manuscrito

No segundo capítulo são explicitadas as características principais do modelo dinâmico do ROV, o métodos adotados para o tratamento da imagem da câmera e o uso do método dos mínimos quadrados para calcular os parâmetros do sistema. No terceiro capítulo são apresentados os dados e os cálculos utilizados para o desenvolvimento e refinamento do controle de profundidade. No quarto capítulo são mostrados os detalhes do programa utilizado para implementar o controle elaborado por [1] e o desenvolvido neste relatório. No quinto capítulo são detalhados os testes realizados e os resultados obtidos. No sexto capítulo conclui-se o trabalho desenvolvido, tendo como base na teoria e os resultados, além de recomendar trabalhos futuros a serem desenvolvidos.

Capítulo 2

Fundamentação Teórica

Para que se possa identificar o sistema a ser controlado, é necessário conhecimento sobre o modelo matemático de veículos submarinos. Apesar de já estar bastante difundido na literatura, este capítulo apresentará um resumo acerca da noção da cinemática e da dinâmica referente aos sistemas subaquáticos.

2.1 Cinemática

O veículo possui seis graus de liberdade sendo três de translação (avanco, deriva e afundamento) e três de rotação (arfagem, guinada e jogo). Para caracterizar esse sistema são utilizados dois referenciais diferentes, um para a terra (fixo, inercial) e um para o veículo (móvel) que normalmente se localiza em seu centro de gravidade [2]. A Figura 2.1 ilustra esses sistemas de coordenadas enquanto que a tabela 2.1 define o que significa cada um dos eixos.

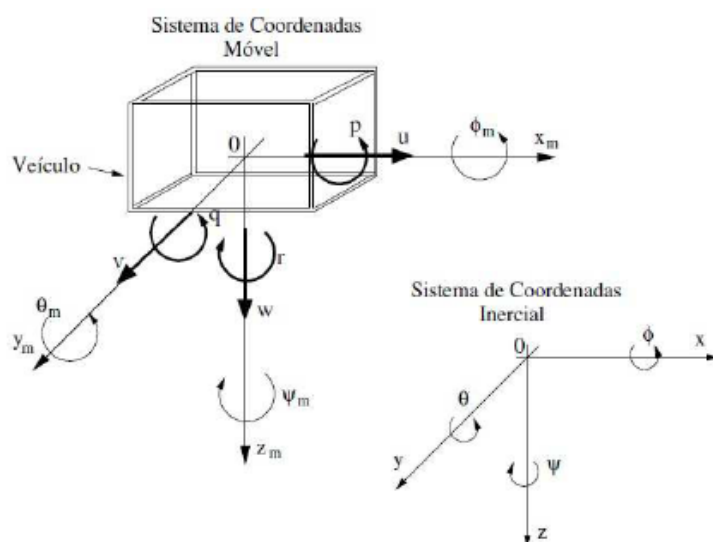


Figura 2.1: Sistemas de Coordenadas

Tabela 2.1: Graus de Liberdade e suas Variáveis				
Graus de Liberdade		Forças e Momentos	Velocidade Linear e Angular	Posição Inercial
1	Movimento na Direção X (avanço) - <i>Surge</i>	X	u	x
2	Movimento na Direção Y (deriva) - <i>Sway</i>	Y	v	y
3	Movimento na Direção Z (afundamento) - <i>Heave</i>	Z	w	z
4	Rotação sobre o eixo X (jogo) - <i>Roll</i>	K	p	ϕ
5	Rotação sobre o eixo Y (arfagem) - <i>Pitch</i>	M	q	θ
6	Rotação sobre o eixo Z (guinada) - <i>Yaw</i>	N	r	ψ

Para ambos os sistemas podem-se representar as grandezas associadas às coordenadas dos veículos. Quando se trata do referencial da terra, pode-se expressar a posição (η_1) e a orientação (η_2) como sendo

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \quad (2.1)$$

De forma análoga, ao representar as velocidades de translação e rotação usando o referencial do corpo, expressa-se o vetor como

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \quad (2.2)$$

Agora, com as notações já definidas, é necessário restringir certos valores pois caso contrário, ocorrerão singularidades na análise dos resultados. Assim, é possível restringir o sistema a partir destas singularidades

$$-90^\circ < \theta < 90^\circ \quad (2.3)$$

$$-180^\circ < \phi, \psi < 180^\circ \quad (2.4)$$

Estabelecidas as notações e consideradas restrições devido às singularidades, representa-se cinematicamente o veículo no espaço.

Mas antes é preciso notar que, derivando as matrizes η_1 e η_2 obtêm-se as velocidades de translação e rotação no referencial da terra. O termo $\dot{\eta}_1$ pode então ser definido por:

$$\dot{\eta}_1 = J_1(\eta_2)v_1 \quad (2.5)$$

O operador de transformação não linear J_1 é dado por uma composição de matrizes de rotação, como visto na Equação (2.6). Tais matrizes podem ser obtidas rotacionando, em um ângulo

qualquer, cada componente de um vetor no sentido anti-horário em um eixo de rotação. Abaixo ilustra-se esse termo com as matrizes representadas em ordem alfabética.

$$J_1(\eta_2) = M_{x,\phi}^T M_{y,\theta}^T M_{z,\psi}^T \quad (2.6)$$

onde as matrizes de rotação são expressas por

$$M_{x,\phi}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (2.7)$$

$$M_{y,\theta}^T = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.8)$$

$$M_{z,\psi}^T = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Com isso, J_1 pode ser simplificado para a seguinte matriz:

$$J_1 = \begin{bmatrix} \cos\theta\cos\phi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\cos\phi & \sin\phi\sin\theta\cos\psi + \cos\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (2.10)$$

Da mesma forma, pode-se obter o termo $\dot{\eta}_2$ e representá-lo como sendo

$$\dot{\eta}_2 = J_2(\eta_2)v_2 \quad (2.11)$$

e também se consegue obter J_2 a partir das matrizes de rotação, usando a seguinte fórmula de [2]

$$v_2 = \begin{bmatrix} \phi \\ 0 \\ 0 \end{bmatrix} + M_{x,\psi} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + M_{y,\theta} M_{x,\phi} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = J_2^{-1}(\eta_2)\dot{\eta}_2 \quad (2.12)$$

e assim se obtêm

$$J_2(\eta_2) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \quad (2.13)$$

Observa-se que a matriz J_2 não está definida para valores de $\theta = \pm 90^\circ$ [2]. Porém tal restrição não se caracteriza como um problema, visto que, como foi especificado na Equação (2.3), o veículo possui uma singularidade nestes pontos.

Outra notação que também será importante está ressaltada na Equação (2.14). Ela representa os esforços externos que também são considerados na modelagem do sistema, onde τ_1 representa o vetor de forças e τ_2 o vetor de momentos.

$$\tau = \begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^T \quad (2.14)$$

2.2 Dinâmica

De acordo com a mecânica clássica, quando uma força é aplicada em um ponto, obtém-se a 2ª lei de Newton,

$$m\ddot{\eta}_1 = \tau_1 \quad (2.15)$$

Entretanto, o ROV estudado não possui dimensões desprezíveis e portanto precisa ser analisado como um corpo rígido. Para efeito de simplificação, assume-se que:

- O corpo possui massa uniforme. Facilitando assim os cálculos para o seu centro de massa;
- A posição do seu centro de massa é invariante;
- O seu centro de gravidade se localiza em uma posição diferente da origem do sistema de coordenadas móvel;

Logo, as equações de translação e rotação da dinâmica do veículo [3] podem ser expressas respectivamente por

$$m[(\dot{\nu}_1 + \nu_2) \times (\nu_1 + \dot{\nu}_2) \times (r_G + \nu_2 \times (\nu_2 + r_G))] = \tau_1 \quad (2.16)$$

$$\frac{d}{dt}(I_0\nu_2) + mr_G \times (\dot{\nu}_1 + \nu_2 \times \nu_1) = \tau_2 \quad (2.17)$$

sendo

$$\frac{d}{dt}(I_0\nu_2) = I_0\dot{\nu}_2 + \nu_2 \times (I_0\nu_2) \quad (2.18)$$

em que r_G representa o vetor distância entre o centro de massa do robô e a origem $O_{x_my_mz_m}$ (referencial móvel). A matriz I_0 reúne momentos e produtos de inércia do robô em relação a $O_{x_my_mz_m}$, e o operador \times representa o produto vetorial.

A partir das equações de translação e rotação, respectivamente, (2.16) e (2.17) podem-se separar os termos relativos ao movimento do centro de massa do corpo (M_{CR}) daqueles que acontecem devido às forças centrípeta e de Coriolis (C_{CR}). Coloca-se na forma matricial a seguir:

$$M_{CR}\dot{\nu} + C_{CR}(\nu)\nu = \tau \quad (2.19)$$

sendo

$$M_{CR}\dot{\nu} = \begin{bmatrix} m\dot{\nu}_1 + m\dot{\nu}_2 \times r_G \\ I_0 m\dot{\nu}_2 + m r_G \times \dot{\nu}_1 \end{bmatrix} \quad (2.20)$$

$$M_{CR} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & mz_G & -my_G & I_x & -I_{xy} & -I_{xz} \\ -mz_G & 0 & mx_G & -I_{yx} & I_y & -I_{yz} \\ my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \quad (2.21)$$

2.2.1 Esforços Hidrodinâmicos

Os esforços hidrodinâmicos podem ocorrer devido a dois fatores: massa adicionada ou ao arrasto hidrodinâmico.

Os esforços devido à massa adicionada estão relacionados com a movimentação do fluido que circunda o veículo e possui uma certa aceleração. São dependentes da profundidade de operação, o que os tornam relevantes para os cálculos utilizados neste trabalho. Podem ser modelados a partir da relação abaixo, de acordo com[3]

$$\tau_A = -M_A\dot{\nu} - C_A(\nu)\nu \quad (2.22)$$

onde M_A representa as derivadas hidrodinâmicas em relação à aceleração do veículo e C_A representa a matriz centrípeta de Coriolis.

Os esforços ocorridos devido ao arrasto hidrodinâmico estão relacionados ao arrasto causado pelo pressão e aos fenômenos de fricção de superfície. São dependentes da forma e da frequência de movimento do veículo, além de apresentarem dependência quadrática em relação à velocidade de deslocamento relativa a correnteza, ν_c que será discutida na sessão de esforços ambientais. Assim calcula-se o esforço de arrasto F_D através da expressão [3]

$$F_D = -0.5\rho C_d S |\nu| \nu \quad (2.23)$$

onde ρ simboliza a massa específica do fluido circundante, C_d uma matriz de coeficientes de arrasto hidrodinâmico e S a área do veículo em contato com fluido. Aproximando essa expressão

por uma estrutura desacoplada, obtém-se

$$F_D(\nu) \approx D(\nu)\nu = C_{dl}\nu + C_{dnl}|\nu|\nu \quad (2.24)$$

em que C_{dl} e C_{dnl} representam matrizes constantes e diagonais com coeficientes de arrasto linear e não linear respectivamente.

2.2.1.1 Forças de Morison

Essas duas componentes dos esforços hidrodinâmicos definem, na literatura, termo *Forças de Morison* [3]. O qual é expresso por:

$$F = \frac{1}{2}C_d\rho S|\nu_c(t) - \nu(t)|(\nu_c(t) - \nu(t)) + C_m\rho V[\dot{\nu}_c(t) - \dot{\nu}(t)] \quad (2.25)$$

onde o primeiro termo da soma é relativo ao arrasto e é desmembrado pela aproximação da Equação (2.24) e o segundo termo se refere à massa adicionada.

Na Equação (2.25) o termo S simboliza uma matriz diagonal, cujos termos são as áreas projetadas em cada eixo e os termos C_d e C_m são matrizes consideradas diagonais (para uma análise simplificada) onde C_m representa o coeficiente de inércia. Esses coeficientes de arrasto e inércia são funções do escoamento, da rugosidade e de características geométricas do corpo.

A natureza do escoamento exerce influência direta sobre a relevância dos componentes linear e quadrática do termo de arrasto. Com um escoamento mais turbulento a componente quadrática se sobressai em relação à linear, enquanto que para escoamentos laminares a componente linear se torna mais importante.

2.2.2 Esforços Ambientais

De acordo com [3] os esforços ambientais são esforços que ocorrem devido à ação da correnteza, que pode ser produzida por diversos fatores naturais como marés, o vento, diferenças de salinidade e convecção. Assim como os esforços hidrodinâmicos que ocorrem devido à massa adicionada, os esforços ambientais possuem uma componente inercial e outra centrípeta, como pode ser visto abaixo

$$\tau_C = -M_C\dot{\nu}_C - C_C(\nu_C)\nu \quad (2.26)$$

Com as matrizes $M_C\dot{\nu}_C$ e $C_C(\nu_C)\nu$ estão representadas abaixo, sendo ρ , a massa específica do fluido circundante, ∇ o volume de fluido deslocado pelo veículo e r_B o vetor distância entre o centro de empuxo e a origem do sistema de coordenadas do veículo $O_{x_my_mz_m}$.

$$M_C\dot{\nu}_C = \begin{bmatrix} \rho \nabla \dot{\nu}_{1c} \\ \rho \nabla r_B \times \dot{\nu}_{1c} \end{bmatrix} + M_A\dot{\nu}_C \quad (2.27)$$

$$C_C(\nu_C)\nu = \begin{bmatrix} \rho \nabla \nu_2 \times \nu_1 \\ \rho \nabla r_B \times (\nu_2 \times \nu_{1c}) \end{bmatrix} + C_A \nu_C \nu \quad (2.28)$$

No primeiro, a frequência de excitação das ondas fazem o corpo oscilar e, por isso, não há ondas incidentes. No segundo caso há ondas incidentes, e é causado pela força do vento, ondas e correntezas, e por isso não foram considerados nesse trabalho.

2.2.3 Esforços Restaurativos

Compreendem os esforços restaurativos como a força peso (P) e o empuxo hidrostático (B). São descritos por

$$P = mg \quad (2.29)$$

$$E = \rho g \nabla \quad (2.30)$$

sendo ρ , a massa específica do fluido e ∇ o volume do fluido deslocado pelo veículo. No referencial do veículo estes esforços são definidos como

$$\tau_G(\nu) = J_1^{-1}(\nu_2) \begin{bmatrix} 0 \\ 0 \\ P \end{bmatrix} \quad (2.31)$$

$$\tau_B(\nu) = J_1^{-1}(\nu_2) \begin{bmatrix} 0 \\ 0 \\ -E \end{bmatrix} \quad (2.32)$$

Com isso, define-se a expressão geral para forças e momentos restaurativos como

$$G(\nu) = - \begin{bmatrix} \tau_G(\nu) + \tau_B(\nu) \\ r_G \times \tau_G(\nu) + r_B \times \tau_B(\nu) \end{bmatrix} \quad (2.33)$$

onde r_B simboliza o vetor distância entre a origem do sistema de coordenadas $O_{x_m y_m z_m}$ e o centro de empuxo e r_G o vetor distância entre essa mesma origem e os centros de massa. O sinal negativo apenas indica que a expressão está posicionada do lado esquerdo da igualdade na formulação geral da segunda lei Newton.

É importante frisar que quando se trata da flutuabilidade de veículos aquáticos a maioria é projetada para possuir uma componente hidrostática maior do que a força peso, com o intuito de garantir a segurança do veículo caso ocorra alguma falha nos propulsores. Dessa maneira, é possível recuperar o veículo facilmente. Porém, no caso do ROV requiere-se um consumo de mais energia para se manter submerso [2].

Além disso, os veículos também são projetados de modo que seus centros de empuxo e de massa estejam alinhados perpendicularmente em relação à superfície, de forma a garantir que as forças restaurativas gerem momentos que contribuam para a estabilidade dos ângulos de rolagem (ϕ , *roll*) e arfagem (θ , *pitch*).

2.2.4 Equação Dinâmica

Um importante ponto em se utilizar as equações dinâmicas no referencial do robô é o fato de os coeficientes (matrizes) de arrasto manterem-se constantes devido a estarem diretamente relacionados à características físicas do veículo.

Com isso é possível escrever a dinâmica do veículo como sendo o sistema de equações

$$M\dot{\nu} + C(\nu)\nu + F_D(\nu) + G(\eta) = \tau \quad (2.34)$$

$$\dot{\nu} = J(\eta)\nu \quad (2.35)$$

onde M representa a soma das matrizes de corpo rígido (M_{CR}) e de massa adicionada (M_A), $C(\nu)$ representa a soma das matrizes centrípeta de Coriolis (C_{CR}) e de massa adicionada (C_A), F_D representa a dissipação hidrodinâmica e $G(\eta)$ representa as forças restauradoras devido ao empuxo e ao peso.

Já τ simboliza os esforços relativos a correnteza, ao cabo e aos propulsores:

$$\tau = \tau_c + \tau_{cabo} + \tau_{prop} \quad (2.36)$$

Como τ_c possui componentes inerciais e de Coriolis, esse termo pode ser decomposto e redistribuído em M e C . Assim, obtém-se a equação final

$$M\dot{\nu} + C(\nu_r)\nu + F_D(\nu_r) + G(\eta) = \tau_{cabo} + \tau_{prop} \quad (2.37)$$

Como este trabalho visa a implementação de um controlador para baixas velocidades e em um ambiente em turbulência, logo pode-se simplificar essa equação aproximando a dissipação hidrodinâmica e as forças restauradores por constantes.

2.3 Método dos Mínimos Quadrados

Para realizar a identificação dos parâmetros da equação que rege a dinâmica do ROV utilizou-se o método dos mínimos quadrados [4]. Tal método consiste em encontrar a função que possua um melhor ajuste considerando o conjunto de dados fornecidos, no atual escopo, a propulsão da turbina (da parte de cima do ROV) e aceleração no eixo z.

Para tal, o método visa minimizar a soma dos quadrados das diferenças entre os valores tabelados Y_i e os valores obtidos pela estimação \hat{Y} como mostrado por

$$\sum_{i=0}^n (Y_i - \hat{Y})^2 = e \quad (2.38)$$

Para uma identificação mais simplificada se fez o uso da segunda lei de newton juntamente com os esforços apresentados neste capítulo para elucidar o somatório de forças que atuam sobre o ROV. Com isso tem-se:

$$m\ddot{Z}_p = E + P + C_1U + \alpha\dot{Z}_p \quad (2.39)$$

Sendo

- m : massa;
- E : Empuxo;
- P : Força peso;
- U : Termo de controle;
- Z_p : Profundidade;
- C_1 : Constante que representa o termo que multiplica os termo de controle;
- α : Constante que simboliza a soma das matrizes de corpo rígido e de massa adicionada.

Considerando o peso e o empuxo constantes durante toda submersão, a critério de simplificação considera-se que a soma dessas forças também será uma constante, C_2 como mostrado em

$$m\ddot{Z}_p = C_2 + C_1U + \alpha\dot{Z}_p \quad (2.40)$$

Dividindo ambos os lados da equação anterior por m , é fácil observar que as constantes serão divididas pela massa. Novamente, a critério de simplificação, considera-se que tal divisão resultará em outra constante, com isso a equação anterior resulta na Equação (2.41) mostrada a seguir.

$$\ddot{Z}_p = C_2 + C_1U + \alpha\dot{Z}_p \quad (2.41)$$

Como \ddot{X} , \dot{X} e U são um conjunto de dados da aceleração, velocidade e propulsão no eixo z, respectivamente, o modelo matemático final da dinâmica do ROV se apresenta da seguinte maneira

$$\hat{\ddot{X}} = \begin{pmatrix} 1 & u_1 & \dot{z}_{p1} \\ \vdots & \vdots & \vdots \\ 1 & u_n & \dot{z}_{pn} \end{pmatrix} \begin{pmatrix} C_2 \\ C_1 \\ \alpha \end{pmatrix} \quad (2.42)$$

Como o erro e esperado para o quadrado da diferença é 0, a fórmula do método pode ser dada como apresentada em (2.43).

$$\sum_{i=0}^n [Y_i - \hat{Y}]^2 = 0 \Rightarrow \sum_{i=0}^n [Y_i - \hat{Y}] = 0 \Rightarrow Y_{0,\dots,n} = \hat{Y}_{0,\dots,n} \quad (2.43)$$

Desta forma, aplicando o método dos mínimos quadrados para os dados e equações do ROV, têm-se

$$\begin{aligned} \ddot{Z}_p &= \hat{\dot{Z}}_p \\ \hat{\dot{Z}}_p &= \begin{pmatrix} 1 & u_1 & \dot{z}_{p1} \\ \vdots & \vdots & \vdots \\ 1 & u_n & \dot{z}_{pn} \end{pmatrix} \begin{pmatrix} C_2 \\ C_1 \\ \alpha \end{pmatrix} \\ \hat{\dot{Z}}_p \begin{pmatrix} 1 & u_1 & \dot{z}_{p1} \\ \vdots & \vdots & \vdots \\ 1 & u_n & \dot{z}_{pn} \end{pmatrix}^{-1} &= \begin{pmatrix} C_2 \\ C_1 \\ \alpha \end{pmatrix} \end{aligned} \quad (2.44)$$

Com os dados da profundidade disponíveis é possível realizar a derivada e a segunda derivada a fim de obter os valores de \dot{x}_0 até \dot{x}_n e de \ddot{Z}_p que juntamente com os da propulsão viabilizam a obtenção dos valores das constantes do modelo matemático.

2.4 Processamento da Imagem

2.4.1 Contextualização do problema

Em uma determinada parte do trabalho, foi necessário implementar o controle de profundidade com os controles de avanço e guinada desenvolvidos por [1]. Porém uma grande dificuldade estava atrapalhando essa implementação conjunta. Enquanto o controle de profundidade demorava para estabilizar, os outros já haviam se tornado instáveis, logo não havia possibilidade de os controles operarem em conjunto. Era necessário refiná-los afim de estabilizá-los.

Esses controles utilizam como referência o tratamento das imagens da câmera do ROV. O controle de deslocamento tem como objetivo manter o cano no meio da câmera enquanto que o de avanço tenta manter o ROV a uma certa distância do cano (neste trabalho foi definida a distância de 30 cm). Para tal uma linha da imagem é binarizada, contendo apenas pixels brancos e pretos, para que o cano seja a referência na linha toda. A Figura 2.2 abaixo identifica a imagem original e sua versão com o tratamento utilizando a linha binarizada.

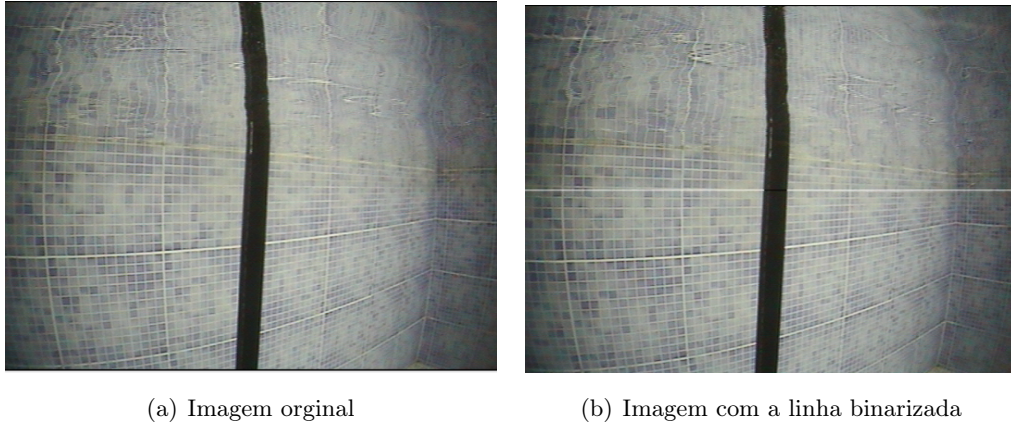


Figura 2.2: Binarização da linha da imagens

Maiores detalhes sobre o processo de binarização da imagem podem ser encontrados em[1].

Depois de repartida a imagem, adquire-se duas faixas de pixels brancos e uma de pixels pretos. O ideal seria que esse padrão se mantivesse. Qualquer variação mínima de luminosidade (sombra ou reflexo) cria um pixel branco no meio da faixa de pixels pretos ou um pixel preto no meio de uma das faixas de pixels brancos, devido a alta sensibilidade do *threshold*, um valor de limiar utilizado para efetuar a binarização da linha. Com isso, o programa identifica mais de uma referência para o controle, que se torna instável.

Assim, uma solução para esse problema seria aplicar os processos de dilatação e erosão de imagem. Isso eliminaria pixels indesejados e daria mais estabilidade ao controle, aumentando a possibilidade de integração com outros.

2.4.2 Morfologia Matemática

A palavra morfologia é um conceito da biologia que estuda as formas e estruturas dos animais e das plantas. Quando se fala em Morfologia Matemática, insere-se essa palavra em um contexto de manipulação de imagem para que se possa ter como saída a estrutura ou forma desejada[5].

A linguagem usada em Morfologia Matemática é a Teoria dos Conjuntos. Ao utilizar uma imagem binária, pode-se dizer que os pixels pretos são o conjunto que descreve completamente essa imagem. Ao analisar uma imagem binária, os conjuntos em questão fazem parte de um espaço bidimensional de números inteiros Z^2 , em que cada elemento do conjunto é um vetor bidimensional e suas coordenadas são dadas pelas coordenadas dos pixels pretos da imagem (x_i, y_i) . Sendo A um conjunto em Z^2 e $a_{n,n}$ sendo um elemento de A . Logo

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (2.45)$$

2.4.2.1 Dilatação

Sendo A e B como conjuntos de Z^2 e \emptyset como o conjunto vazio. A dilatação de A por B é dada por [5]

$$A \oplus B = \left\{ z_i | (\hat{B})_x \cap A \neq \emptyset \right\} \quad (2.46)$$

Onde \hat{B} significa a reflexão de B . Esse processo começa na reflexão de B em torno de sua origem e é seguido pela translação dessa reflexão por x_i . Logo, o conjunto de todos esses deslocamentos de x_i tais que A sobreponha pelo menos um elemento não nulo é o que denominado dilatação de A por B . Exemplificado na figura abaixo.

Quando se trata deste trabalho, pode-se usar esse processo para expandir os pixels desejáveis da linha binarizada, de uma maneira que os indesejados são eliminados.

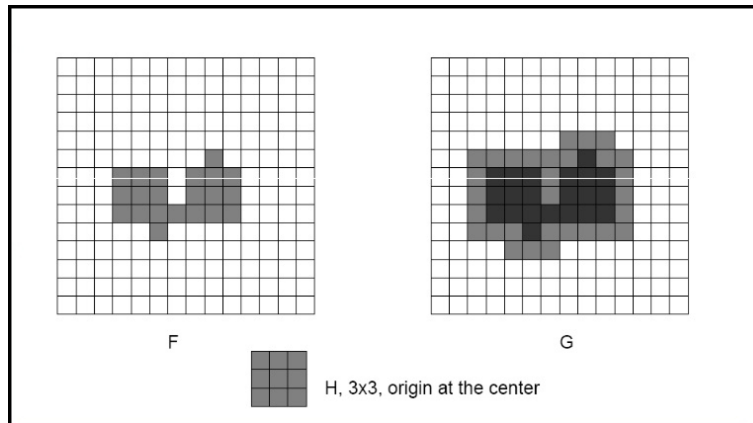


Figura 2.3: Exemplo de Dilatação de Imagem

2.4.2.2 Erosão

Agora também tem-se A e B como conjuntos de Z^2 . A erosão de A por B é dada por [5]

$$A \ominus B = \{x_i | (B)_x \subseteq A\} \quad (2.47)$$

A erosão de A por B é o conjunto de todos os pontos x_i tais que B , quando transladados por x_i , fique contido em A . Isso quer dizer que a erosão é a operação dual da dilatação, ou seja, a erosão de A por B também significa a dilatação de B por A . Podemos ver um exemplo abaixo.

Neste trabalho, usa-se a erosão para eliminar os pixels extras que foram criados a partir da dilatação dos pixels desejados. O processo funciona como um filtro. A dilatação elimina os pixels indesejados aumentando o número de desejados. Porém isso cria pixels extras que também precisam sumir. Assim usa-se a erosão para filtrar os últimos pixels indesejados.

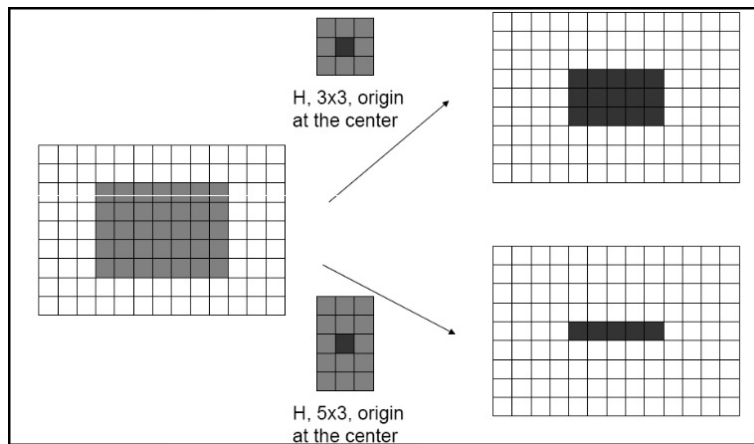


Figura 2.4: Exemplo de Erosão de Imagem

Capítulo 3

Implementação Computacional

Nesta parte do relatório, será especificado o programa utilizado para controlar o ROV. O programa roda na plataforma Visual Studio no Windows sendo a linguagem C#, a linguagem de programação utilizada. O Nome do programa é *vrSimpleSample*, levando esse nome pois foi concebido a partir do programa base, fornecido pela *VideoRay*. Ao rodar o programa, é possível ver muitos arquivos que o compõem. O arquivo principal é o *Form1.cs* e outro de igual importância é o *Form1.Designer.cs*, Figura 3.1. O primeiro é onde está escrito o código principal, enquanto o segundo é o programa que desenha a interface que interage com o usuário.

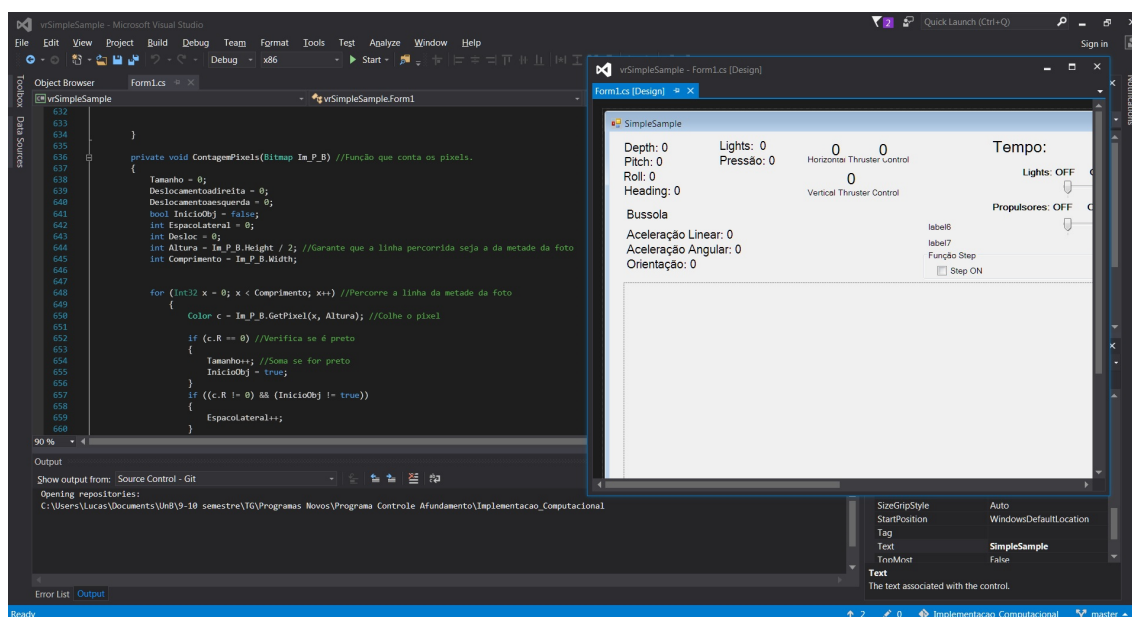


Figura 3.1: Arquivos Form.cs e Form.Designer.cs no Visual Studio

Para detalhar as funções presentes no código principal, primeiro, é necessário saber que todas as partes que estão comentadas em inglês fazem parte do programa base que foi recebido e, portanto, não foram alteradas e não serão comentadas aqui.

3.1 Form1

Função principal do programa. Ela inicializa todas as variáveis dos sensores e também inicializa a câmera. Após todas essas inicializações, entra-se na parte lock, recurso disponível em C# para sincronizar a utilização de recursos.

Primeiramente percebe-se a inicialização das variáveis de propulsão. Depois são definidos os labels que aparecem na interface quando o programa inicializa. A seguir existe uma variável booleana que define se a função *AnotaDados* será utilizada e logo após define-se a contagem de tempo que foi utilizada. Percebe-se que o programa é atualizado a cada 0.1 segundo. Após algumas outras inicializações, nota-se uma parte grande dedicada a função *step*. Essa parte foi uma tentativa fracassada de controle e não será detalhada nesse relatório.

Percebe-se também que essa função chama outras funções como *ContPix*, *ControleProfundidade*, *ControleDesloc* e *ControleDista*. A seguir, nota-se uma sequência de “ifs” de grande importância. Eles definem como as variáveis de controle serão aplicadas aos propulsores. É necessário definir um limite máximo de valor passado a propulsão para que o valor não seja muito alto e não corra o risco de danificar o ROV.

Por último, definem-se condições para a ativação da função *AnotaDados* e do acionamento das luzes.

3.2 button1_click e Luzes_scroll

Duas funções que controlam as luzes do ROV. A primeira define a variável booleana como true quando o botão é acionado. A segunda diz que as luzes devem ser ligadas se a variável booleana for true e desligadas caso seja false.

3.3 button1_click1

Destinado ao botão STOP. Para o programa, desliga as luzes caso estejam ligadas e desliga os propulsores caso estejam ligados.

3.4 trackBar1_Scroll

Define qual variável está associada a qual ponto da *trackBar* Propulsores. Se estiver acionado define a variável ControlandoROV como true e se estiver desativada define essa variável como false.

3.5 button1_Click_2

Destinado ao botão SNAPSHOT, que salva as imagens da câmera.

3.6 Video_Click

Destinado ao botão VIDEO, que salva as imagens da câmera em sequência, para se tornar um vídeo.

3.7 AnotaDados

Função que lê os dados dos sensores do ROV e os escreve em arquivos .txt, cada um contendo os dados recebidos a cada 0.1 segundo, depois de iniciado o programa.

3.8 GrayScaleFilter

Essa função é destinada a contar os pixels brancos e pretos para descobrir parâmetros e assim poder aplicar o controle de deslocamento e distância.

Primeiro nota-se que a linha binarizada será composta por um número N_{pb} de pixels brancos, seguido por um número N_{pp} de pixels pretos e por fim, de um número N_{pb2} de pixels brancos, que segue a regra:

$$Desloc = \frac{|N_{pb2} - N_{pb}|}{2} \quad (3.1)$$

Essa contagem de pixels é depois convertida para centímetros e enviada para a função *ControleDesloc*, para posteriormente chamar a função *Pix_To_Cm* no intuito de descobrir o valor que será enviado ao *ControleDista*.

Depois, os pixels processados passam pelos processos de dilatação e erosão de imagem, que foi explicado no capítulo 2. Aqui, os pixels que serão processados são os brancos. Para se obter um melhor resultado final, os pixels brancos passam primeiro pelo processo de erosão com o intuito de aumentar o tamanho do cano. Depois aplica-se duas vezes o processo de dilatação para que os poucos pixels pretos que estão no meio da linha branca desapareçam. Com os seguidos processos de dilatação, o cano (pixels pretos) aparenta diminuir de tamanho, logo é preciso um último processo de erosão para que ele volte ao tamanho normal.

3.9 ContPix

Função que chama as funções *GrayScaleFilter* e *ContagemPixels*. Ela é chamada pela função principal *Form1*.

3.10 Pix_To_Cm

É uma função chamada pela função *ContagemPixels*, e usa uma fórmula calculada a partir de testes encontrados no trabalho [1] para conseguir fazer uma conversão do número de pixels para centímetro.

3.11 ControleProfundidade

Esta função utiliza os ganhos calculados no Capítulo 4, para implementar o controle PID da profundidade e seus refinamentos, enviando um comando (F6) para o propulsor situado na parte de cima do ROV.

3.12 ControleDesloc

Função que determina o comando do controle (F3) que será passado aos propulsores de forma que o cano permaneça no centro da câmera.

A função usa as parâmetros calculados e define as ganhos proporcional, derivativo e integral que definirão o controle PID que atuará nos propulsores. O controle é atualizado a cada 0.1 segundo. Foi criado um filtro que, a partir de tentativa e erro, definiu-se um novo comando dos propulsores que só utilizará 20% do novo controle enquanto os 80% restantes serão do antigo. Isto evita uma mudança brusca nos valores passados ao propulsor, evitando possíveis danos ao ROV.

3.13 ControleDista

Função que determina o controle que será passado aos propulsores (F4) de forma que o cano permaneça sempre a uma certa distância do ROV. Mesma lógica de *ControleDesloc*.

Capítulo 4

Desenvolvimento do Sistema de Controle

Antes de iniciar os cálculos e a elaboração do controle de profundidade do ROV, foi necessário a criação de um script no *Matlab* para utilizar os dados, fornecidos pela função *AnotaDados* e disponíveis em formato *.txt*, na construção de gráficos e na obtenção dos coeficientes citados na Capítulo 3, e por fim, realizar os cálculos dos ganhos do controlador PID. Foram elaborados manualmente vários sinais de propulsão, com alcance máximo de -100 a 100, sendo -100 a maior rotação no sentido anti-horário e 100, no sentido horário, para gerar os conjuntos de dados utilizados na identificação.

Através dos dados de tempo decorrido, profundidade e propulsão do ROV, foi possível obter através do script os gráficos das Figuras 4.1 e 4.2.

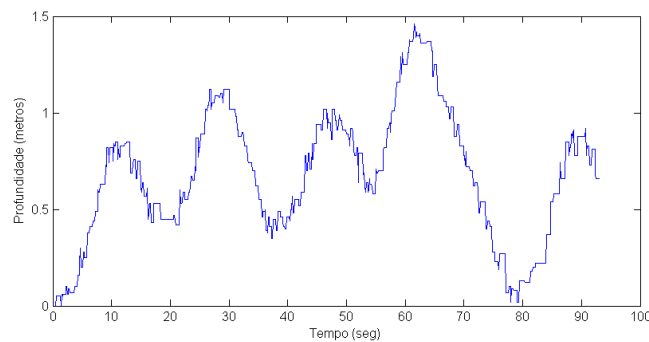


Figura 4.1: Evolução Temporal da Profundidade.

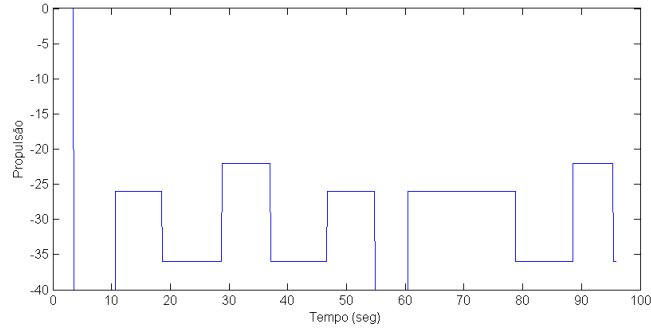


Figura 4.2: Evolução Temporal da propulsão.

No mesmo script também foi possível obter os gráficos da aceleração e velocidade, respectivamente, Figuras 4.3 e 4.4, derivando os dados de profundidade.

A derivada utilizada para a obtenção dos valores da velocidade e aceleração não foi baseada na fórmula padrão e sim na Equação (4.1), tendo em vista que, devido à natureza ruidosa dos dados de profundidade, tornou-se necessário também uma certa filtragem de tais dados.

$$\dot{x}(i) = P_1 \frac{x(i+1) - x(i-1)}{2\Delta t} + \dots + P_n \frac{x(i+n) - x(i-n)}{(n+1)\Delta t} \quad (4.1)$$

Sendo n o tamanho da "janela" de dados utilizada para a filtragem e P o peso atribuído para o cálculo da derivada de cada conjunto de pontos.

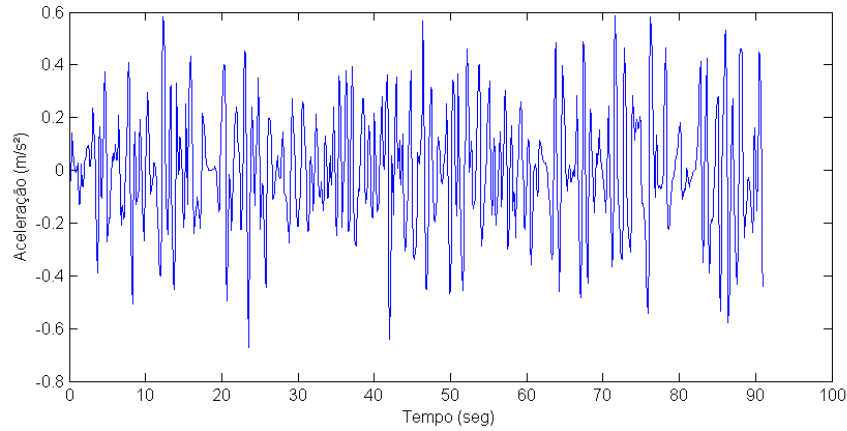


Figura 4.3: Gráfico Evolução Temporal da Aceleração.

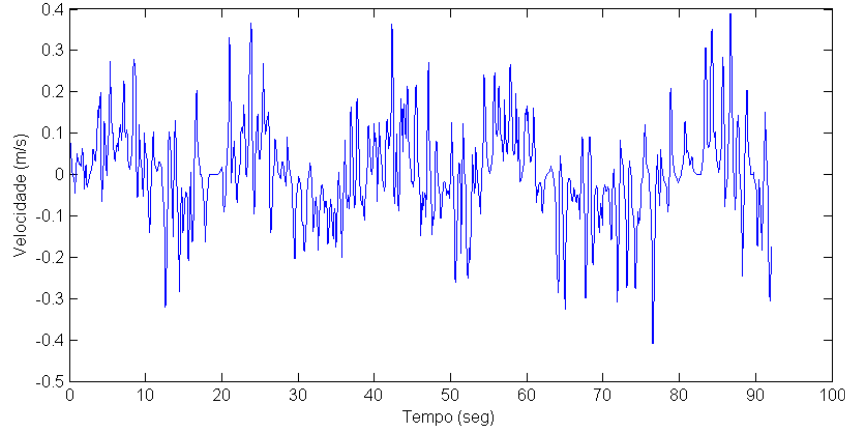


Figura 4.4: Evolução Temporal da velocidade.

4.1 Controle PID

O controle implementado em um primeiro momento foi o PID. Porém antes de realizar os devidos cálculos, foi necessário linearizar a Equação (2.41) em torno de um ponto de operação. Isso foi preciso pois o ROV necessita de uma força mínima para que ele fique imóvel enquanto estiver submerso. Logo a entrada U (propulsão) assumiu a forma mostrada na equação (4.2) a seguir.

$$U = U' - \frac{C_2}{C_1} \quad (4.2)$$

De forma que a equação linearizada apareça como mostrada na equação (4.3).

$$\hat{\ddot{X}} = C_1 U' + \alpha \dot{X} \quad (4.3)$$

Para elaborar a equação característica desejada no controle PID, chegou-se a conclusão que alocando três polos reais e de mesmo valor, poder-se-ia obter a melhor relação entre tempo de resposta e possibilidade de implementação, além disso o sistema não iria oscilar[1]. Desta forma a equação característica se caracteriza como

$$(s + a)^3 = s^3 + 3a^2s + 3as^2 + a^3 \quad (4.4)$$

Logo o sistema desejado seria então

$$\frac{b}{U_{ref}} = \frac{K(s)}{s^3 + 3a^2s + 3as^2 + a^3} \quad (4.5)$$

$$K(s)U_{ref} = bs^3 + 3a^2bs + 3abs^2 + ba^3 \quad (4.6)$$

Para relacionar as Equações (4.3) com a (4.6), adotou-se $\hat{X} = bs$, resultando na equação (4.7) em seguida.

$$K(s)U_{ref} = s^2\hat{X} + 3a^2\hat{X} + 3as\hat{X} + \frac{a^3\hat{X}}{s}$$

$$K(s)U_{ref} = \hat{\ddot{X}} + 3a^2\hat{\dot{X}} + 3a\hat{\ddot{X}} + a^3 \int \hat{X} \quad (4.7)$$

Para realizar o controle PID é sabido que para os cálculos do canal proporcional e integral é necessário o valor referencial[2], logo $K(s)$ teria a forma de

$$K(s) = 3a^2 + \frac{a^3}{s} \quad (4.8)$$

Utilizando as equações (4.8), (4.7) e (4.3), o comando de controle se configura da seguinte maneira

$$U' = -\frac{3a^2}{C_1}(\hat{X} - U_{ref}) - \frac{3a - \alpha}{C_1}\hat{\dot{X}} - \frac{a^3}{C_1} \left(\int \hat{X} - \int U_{ref} \right) \quad (4.9)$$

onde $\hat{X} - U_{ref}$ simboliza erro entre o valor da referência e o valor real. Com isso, os ganhos do controlador PID são

$$K_P = -\frac{3a^2}{C_1} \quad (4.10)$$

$$K_I = -\frac{a^3}{C_1} \quad (4.11)$$

$$K_D = -\frac{3a - \alpha}{C_1} \quad (4.12)$$

4.2 Implementação

Sabe-se que um comando positivo no propulsor vertical significa efetuar um movimento em direção à superfície. Logo, um comando negativo efetuará uma submersão.

Para aplicar o controle obtido, em linguagem C, é necessário adicionar alguns detalhes ao programa. A partir da equação (2.44) descobrem-se os parâmetros que foram utilizados para calcular os ganhos do controle. Como foi explicado, esses parâmetros ajudam a linearizar o controle em volta do ponto de operação, ou seja, o ponto em que a velocidade do ROV é zero ($\dot{X} = 0$). Assim este ponto é dado por

$$U_0 = -\frac{C_2}{C_1} \quad (4.13)$$

Isto significa que, ao atingir o valor de referência, é esperado que o ROV se mantenha parado embaixo d'água com o termo de controle definido como o valor acima. Dessa maneira, para que o ROV comece a se movimentar para baixo, a todo o valor de comando gerado pelo controle é necessário a adição do termo U_0 . Com isso obtêm-se a já mostrada equação (4.2).

4.2.1 Cálculo dos Ganhos

Depois de vários testes, os valores encontrados para C_1 , C_2 e α são

$$\begin{pmatrix} C_1 \\ C_2 \\ \alpha \end{pmatrix} = \begin{pmatrix} -0.0019 \\ -0.0584 \\ -0.0232 \end{pmatrix} \quad (4.14)$$

Sabe-se que os valores de C_1 e C_2 foram usados para se obter o ponto de operação do controle. Os valores de C_1 e α foram usados para o cálculo dos ganhos do controle. Para se ter um valor ideal para a em termos de controle, é necessário relacionar o parâmetro $3a$ do termo bs^2 da Equação (4.6) com um valor proporcional a α . Porém buscar um valor proporcional próximo a α resultou em valores inadequados para que o controle funcionasse. Através de uma validação empírica, um valor mais alto para a fora utilizado para que pudesse realmente gerar um controle de melhor qualidade. O valor escolhido foi de

$$a = 8 \quad (4.15)$$

Com isso, os valores dos ganhos encontrados a partir das Equações (4.10), (4.11) e (4.12) são

$$\begin{aligned} K_p &= 8,8387 \\ K_i &= 0.2210 \\ K_d &= 105.7226 \end{aligned} \quad (4.16)$$

4.3 Problemas encontrados

Muitos problemas foram identificados quando se testava tanto unicamente o controle de profundidade, quanto todos os controles integrados. Cada um deles envolveu o desenvolvimento de diferentes técnicas para sua respectiva solução.

4.3.1 Saturação

Nos testes iniciais, um valor fixo de profundidade era proporcionado como referência diretamente no programa. Dessa maneira, se a referência estiver de certo modo, distante do ponto atual, o erro presente no controle será elevado acarretando em um termo integral muito alto, tal ocorrência se caracteriza como *wind-up* da ação integral e gera um comando de controle saturado.

Uma das soluções adotadas foi fazer um planejamento de trajetória em que se fornece uma referência que aumenta gradativamente ao longo do tempo. Assim, o comando do controle também aumentaria com o tempo, o que geraria uma transição mais suave, um sobressinal menor e a exclusão do efeito *wind-up* da ação integral.

Definiu-se a referência R , em centímetros, como

$$R = 0.1 + 0.01t \quad (4.17)$$

onde t é o contador de tempo.

Além disso foram realizados experimentos para se descobrir em qual faixa de valores o ROV consegue submergir emergir de maneira suave. Foram descobertos que para emergir de maneira suave com o valor máximo de -27 de comando enquanto que para submergir, o valor mínimo é de -39. Portanto o comando deve variar apenas entre esses dois valores. Como medida de proteção aos propulsores do ROV, um valor mínimo de -42 fora estabelecido como saturador do comando do controle de profundidade.

4.3.2 Alta variação do comando

Um dos primeiros problemas notados foi a saturação do comando. No painel do usuário, em que o programa mostra as imagens da câmera e os valores de vários sensores, também é mostrado o valor do comando passado ao propulsor vertical. Quando presente na água o ROV apresenta medições de profundidade com elevada variação, até mesmo se o ROV se encontra parado em um mesmo lugar. Logo se percebe que, por causa dessas variações, o comando do controle também apresenta variação, não se mostrando muito eficaz.

Os controles desenvolvidos por [1] já possuíam um filtro que limitava a variação do comando passado à turbina. Para implementar esse filtro, foi necessário salvar o valor do comando calculado para então ser usado no cálculo do novo comando. Nos controles de [1], era adicionado somente 20% do novo valor de comando ao comando antigo. No programa em C#, esse filtro se aplica seguindo a seguinte fórmula

$$Comando = ComandoAnterior \cdot 0.8 + NovoComando \cdot 0.2 \quad (4.18)$$

Quando o mesmo filtro fora aplicado ao controle de profundidade notou-se que 20% ainda era uma porcentagem muito alta, principalmente pelo fato do controle vertical presar por uma abordagem menos brusca nos movimentos e não exigir uma resposta tão imediata. Logo, com o decorrer de alguns testes, foi definido que a porcentagem do valor fornecido do novo comando seria apenas de 5%. Isso deixa a resposta mais lenta porém muito mais suave. Assim, a nova fórmula se torna

$$Comando = ComandoAnterior \cdot 0.95 + NovoComando \cdot 0.05 \quad (4.19)$$

Além disso, o controle de profundidade requer ainda mais atenção pois, diferente dos controles de deslocamento e distância, sempre há a força do empuxo atuando sobre ele. Isso significa que o ROV possui o valor mínimo para começar a submergir, como já foi discutido na Seção 4.2. Portanto não é prático fornecer valores que não se encontram próximos ao valor inicial U_0 . Então foi necessário aplicar um outro filtro no intuito de manter a faixa de valores em que o ROV atuaria.

4.3.3 Problema com o Ganho Derivativo

Ao analisar os ganhos calculados, Equação (4.16), nota-se que o K_D é muito maior se comparado aos outros e conseqüentemente amplificando quaisquer ruídos e variações oriundos dos sensores de medição. Considerando o ambiente onde o ROV está inserido, tal constatação é de extrema importância pois nele encontram-se diversas perturbações que acarretam em variações e ruídos na medição e conseqüentemente, ganhos derivativos discrepantes foram identificados.

Uma solução adotada para este problema foi a de simplesmente retirar o ganho derivativo do cálculo do controle. Desta forma o comando do controle é

$$U' = K_P \cdot Erro + K_I \cdot \int Erro \quad (4.20)$$

No entanto, sem o ganho derivativo o ROV consome mais tempo para se estabilizar na referência fornecida, pois o próprio auxilia na atenuação da ação integral quando esta se encontra acumulada. Entretanto como a referência fornecida aumenta com o tempo, acarretando em uma transição mais suave, os problemas acerca do sobressinal e do tempo de estabilização do controle são minimizados.

Outra solução adotada foi o desenvolvimento de um filtro para atenuar as variações encontradas no erro. Fazendo-se o uso da função erro de Gauss caracterizada como

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (4.21)$$

O erro filtrado é calculado da seguinte forma

$$\Delta_1 = Erro - ErroAnterior \quad (4.22)$$

$$\Delta_2 = erf\left(\frac{\Delta_1}{V_m}\right) \cdot \Delta_1 \quad (4.23)$$

$$Errofiltrado = ErroAnterior + \Delta_2 \quad (4.24)$$

sendo V_m a variação encontrada nas medidas do sensor. Os resultados do controle PID utilizando tal filtro são mostrados posteriormente neste relatório.

Capítulo 5

Testes e Resultados

5.1 Testes iniciais para conhecimento básico do ROV

Em um primeiro teste com o ROV na água, descobriu-se que a chave ON/OFF dos propulsores da janela do programa *SimpleSample* modificado estava desativada. É de conhecimento prévio que ligar os propulsores fora da água pode danificá-los. Por isso, essa chave estava desativada direto no programa, para que o próximo usuário não a ativasse por engano. Assim, foi necessário uma análise mais a fundo do programa, afim de entender variáveis e conseguir implementar a propulsão.

No segundo dia de teste, procurou-se entender como utilizar a função *AnotaDados*, para plotar os dados obtidos no *MatLab* e obter gráficos referentes aos testes. Depois de uma análise mais detalhada do programa, descobriu-se uma variável booleana, *Anota_txt*, que sinalizava a obtenção ou não de dados pelo programa.

Depois de estudar mais a fundo, muito sobre o programa fora esclarecido. Descobriu-se as variáveis que simbolizavam o controle de deslocamento (F3) e de distância (F4) e a existência de um *threshold* para obter os pixels pretos, necessários para o processamento de imagem e controle.

Foi identificado também que as variáveis, *PropHori* e *PropHori2*, simbolizam respectivamente as turbinas esquerda e direita. Dessa forma os controles desenvolvidos por [1] foram testados, constatando então o sucesso na implementação do controle de deslocamento, embora a turbina esquerda não estivesse respondendo aos comandos de uma forma tão eficaz quanto a direita. Posteriormente, fora gravado um vídeo mostrando o funcionamento do controle. Já o controle da distância não funcionou e novas dúvidas surgiram.

No teste a seco realizado, fora explicitado os valores de F3 e F4 no painel do programa através da criação de duas novas labels e foi compreendido que, para a realização do controle de distância e consequentemente o cálculo do comando F4, é necessário iniciar o programa com a referência já situada no campo de visão da câmera do ROV. Caso contrário o resultado obtido para F4 é *NaN* (Not a Number), resolvendo o problema do controle da distância mostrado anteriormente.

5.2 Ponto de linearização e teste de identificação

Após os testes iniciais, fora dedicado o restante do tempo para testar o propulsor vertical e descobrir quais os valores que faziam o ROV vencer a força empuxo. Primeiro concluiu-se que valores positivos impulsionam o ROV para cima, e negativos para baixo. Após a utilização de diversos valores, o ROV conseguiu vencer a força empuxo a partir do valor -36, submergindo apenas alguns centímetros. Valores menores que -36 promovem uma submersão constante do ROV.

Com a descoberta desse ponto, foi feita a identificação do sistema fazendo variando a profundidade do ROV dentro da piscina, de forma que o ROV submerja sem atingir o fundo da piscina e emerja sem atingir a superfície, relacionando a propulsão da turbina vertical com a distância percorrida ao longo do eixo z . Neste caso foi necessário uma providência para que o ROV não colidisse com as paredes da piscina, caso contrário não seria garantida uma identificação precisa do sistema, assim ele foi posicionado no meio da piscina, antes do início dos testes.

Após obter os dados, oriundos da função *AnotaDados*, e plotar um gráfico da profundidade, Figura 4.1, foram calculadas a primeira e a segunda derivada, respectivamente, Figuras 4.4 e 4.3. Esses foram os dados utilizados para a identificação e controle do sistema, mostrados nos Capítulos 2 e 4.

5.3 Resultados

5.3.1 Primeiro controle de profundidade

Após a utilização dos dados obtidos na identificação, calcularam-se as constantes do sistema, Equação (4.14), e os ganhos do controlador PID, Equação (4.16). Com isso foi possível testar o controlador na prática, utilizando uma referência situada a 80 centímetros abaixo da superfície da piscina.

Os resultados do teste se encontram abaixo. A Figura 5.1 fornece o sinal do controle enquanto a Figura 5.2 representa a saída do sistema controlado.

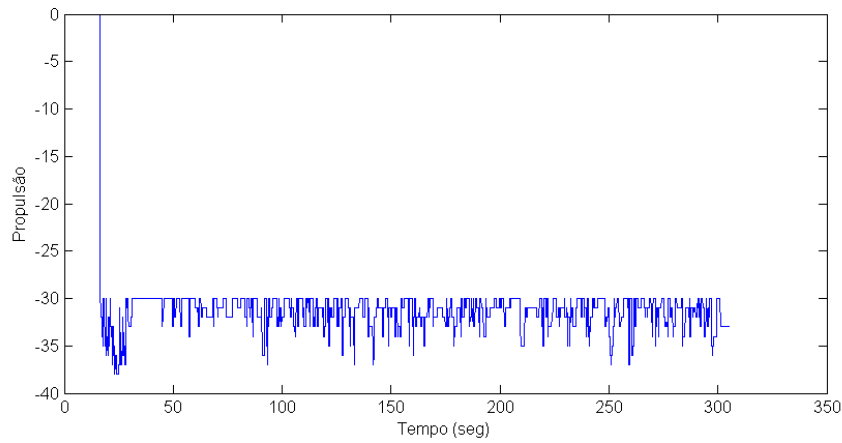


Figura 5.1: Gráfico Propulsão Vertical x Tempo.

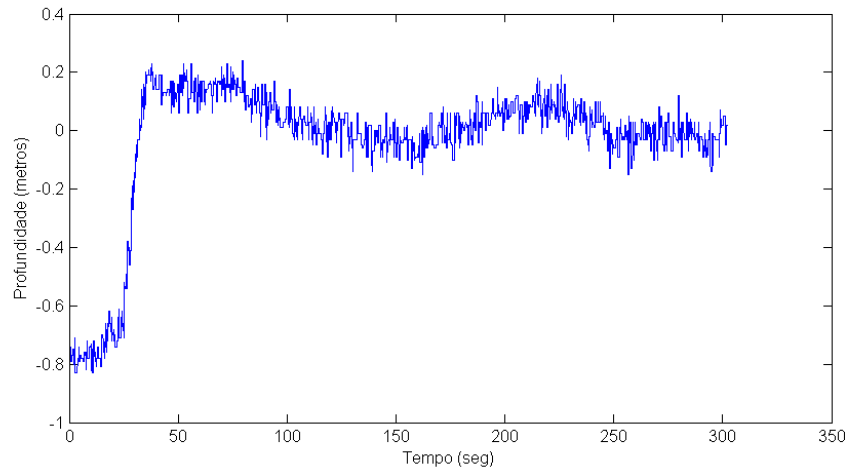


Figura 5.2: Gráfico Profundidade controlada x Tempo.

Analisando a Figura 5.2 é possível perceber que o ROV atinge a profundidade utilizada como referência para o controlador. Ao mesmo tempo observa-se uma oscilação em torno desta profundidade, tal fato pode ser explicado pela presença de ruído no sensor durante a medição da profundidade.

5.3.2 Primeiro controle multivariável

O sucesso do sexto teste possibilitou a realização de um novo experimento, integrando o controle de profundidade com os de distância e deslocamento. Os controles de deslocamento e distância se mostraram eficazes e o resultado do controle de profundidade é ilustrado na Figura 5.3.

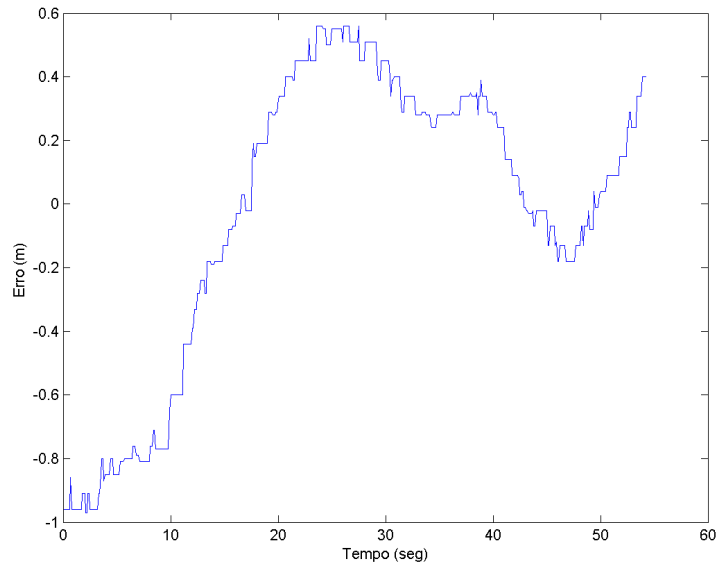


Figura 5.3: Gráfico Erro x Tempo.

A análise da Figura 5.3 indica uma instabilidade no controle de profundidade ao atuar em conjunto com os outros dois controles. A partir disso uma otimização deste controle se fez necessária.

5.3.3 Controle de profundidade com melhorias

Após diversas tentativas observou-se que a parte derivativa do controle PID instabilizava o sinal de controle, devido ao fato desta ser mais sensível as variações do sensor de profundidade. Com isso, implementou-se somente um controle PI, com os mesmos ganhos calculados na Equação (4.16). A Figura 5.4 ilustra o resultado desse controle.

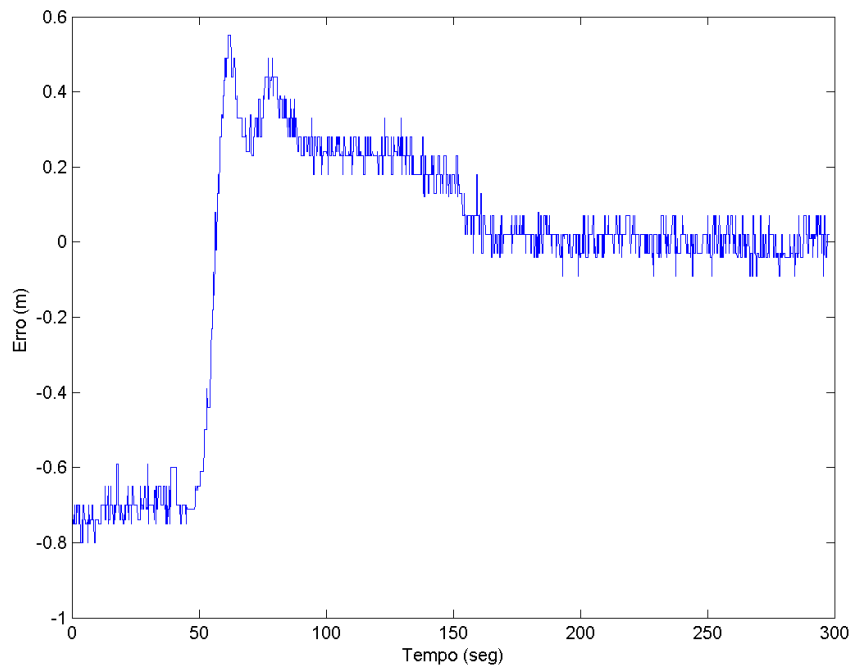


Figura 5.4: Resposta controle PI.

Além disso, notou-se uma maior estabilidade do sinal de controle, Figura 5.5.

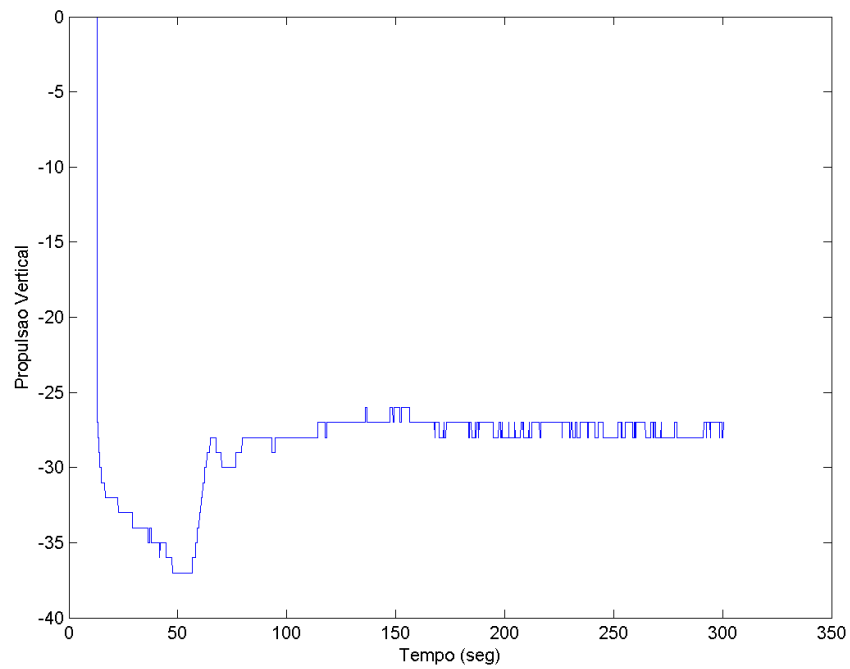


Figura 5.5: Saída do controlador PI.

Outra implementação realizada neste teste fora a do controle PID com a utilização do filtro

para a variação, mostrado nas Equações (4.21) a (4.24). O resultado fornecido por este controle está ilustrado a Figura 5.6, a seguir.

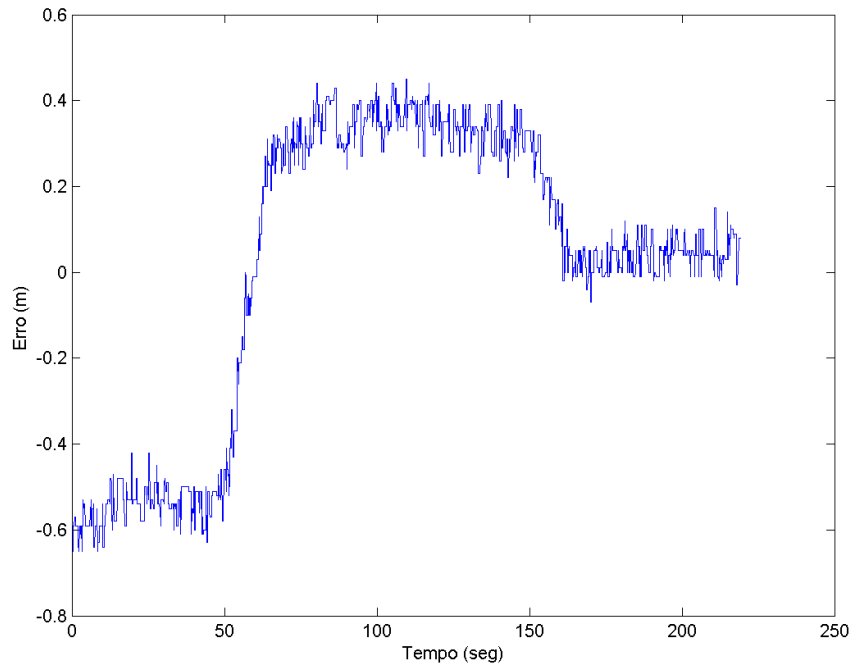


Figura 5.6: Resposta do controlador PID com filtro de erro.

O resultado obtido do controle PID com o filtro para a variação do erro mostra uma melhora significativa nos valores de sobressinal se comparado ao controle PI, mantendo-se também o tempo assentamento em torno dos 150 segundos.

5.3.4 Implementação final do controle multivariável

Realizado o aprimoramento do controle de profundidade, buscou-se uma melhora para os controles de deslocamento e distância. Uma forma encontrada foi a de aplicar os processos de dilatação e erosão no tratamento da imagem oriunda da câmera do ROV. Uma comparação entre a imagem tratada e não tratada pode ser vista na Figura 5.7 a seguir.

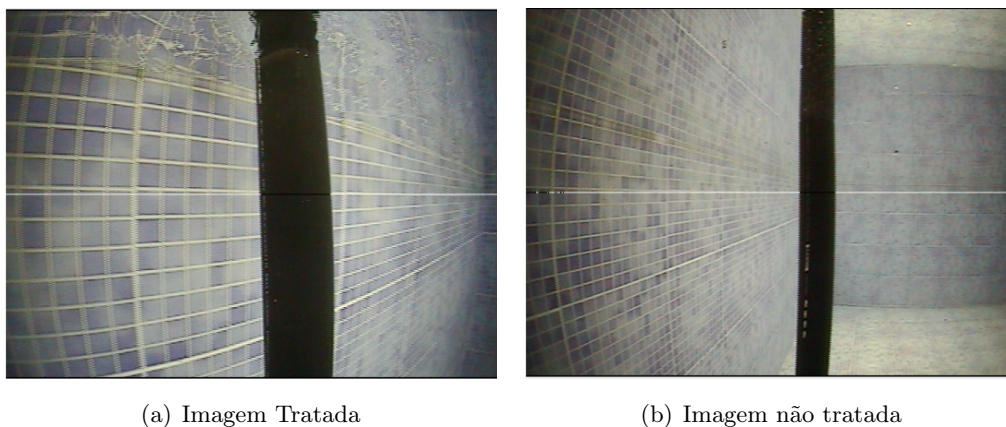


Figura 5.7: Comparação entre as imagens binarizadas

Percebe-se que na imagem tratada apenas o cano apresenta pixels pretos. Assim os controles de deslocamento e distância se tornarão menos instáveis, como foi explicitado na Seção 2.3 .

Ao testar as novas implementações dos respectivos controles, em um experimento com os controladores integrados, percebeu-se que a atuação do controle de profundidade se sobressaía em relação aos outros, deslocando o ROV no sentido horário antes mesmo que o controle de deslocamento pudesse corrigir este erro. Foi necessário a adição de uma compensação a este controle, proporcional ao de profundidade.

Com o devido aprimoramento de todos os controles, tentou-se repetir o sétimo teste. A Figura 5.8 ilustra o resultado obtido.

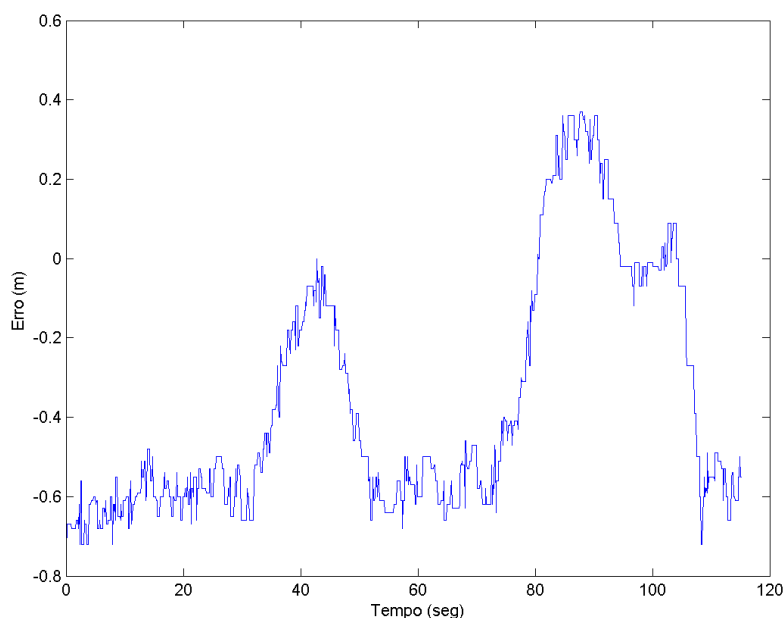


Figura 5.8: Resposta do controlador PID integrado.

Semelhante ao sétimo teste, a integração de todos os controles desenvolvidos não apresentou os resultados esperados. Diversos fatores podem ter contribuído para o resultado encontrado, dentre eles, pode-se citar que os controles foram realizados apenas para atuarem em seus respectivos eixos, não considerando o quanto eles influenciavam os movimentos nos demais. Além disso, os tratamentos realizados adicionaram um maior tempo de processamento para a realização do controle podendo ter influenciado no tempo de resposta.

Outro motivo que pode ter representado um impacto negativo nos resultados é o fato do modelo não considerar as forças e reações dos propulsores. A interação entre o fluxo desses motores podem gerar uma turbulência, o que dificulta a atuação do controle. Um modelo que considere esses fatores se faz necessário para se obter melhores resultados.

Capítulo 6

Conclusões

O objetivo principal do projeto foi implementar um controle multivariável em ROV. A oportunidade de tratar um problema real foi um dos grandes motivos para que este projeto fosse escolhido como trabalho final de graduação. Na prática, há muitos fatores que devem ser considerados que na teoria não são considerados. Muitos projetos envolvem apenas simulações, então sair da zona de conforto foi um grande, porém satisfatório, desafio.

Em primeira mão, foi necessário um maior entendimento do programa em C# com os controles de deslocamento e distância já desenvolvidos. Depois de confirmar que seus cálculos estavam corretos e seus controles funcionavam, foram documentadas todas as funções usadas no programa, afim de facilitar o entendimento de futuros alunos que viessem a trabalhar com o ROV.

Então voltou-se a atenção para o controle de profundidade. Primeiro foi necessário fazer a identificação do sistema a partir de vários testes que puderam elucidar seu comportamento. Com isso tornaram-se possíveis os cálculos dos ganhos para uma posterior implementação de um controle PID.

Nesta etapa foram encontrados diversos problemas especificados neste relatório, desde ruídos presentes nos sensores até a saturação do comando enviado ao propulsor vertical. Foram tomadas medidas específicas para cada caso com o objetivo de otimizar cada um dos controles trabalhados.

Com o refinamento dos controles, voltou-se para a tentativa de implementação conjunta de todos eles. Porém, como cada um foi implementado de forma a não considerar a influência dos outros controles em seu eixo de atuação, o resultado obtido não se mostrou satisfatório, mesmo aplicando compensações proporcionais aos comandos dos outros controles.

Como neste trabalho ficou evidenciado que é possível controlar os eixos de avanço(x), guinada(ψ) e afundamento(z), um próximo passo seria realizar uma identificação conjunta dos eixos citados para posteriormente desenvolver apenas um controle que atuasse em todos eixos ao mesmo tempo. Recomenda-se estudar estratégias de controle não-linear para reduzir algumas hipóteses e chegar a um resultado interessante.

Outros refinamentos que podem ser aplicados no futuro são: desenvolver técnicas que possibilitem um melhor tratamento de imagem, otimizar o programa em C# de modo a evitar erros causados por funções internas criadas pela VideoRay

- Desenvolver técnicas que possibilitem um melhor tratamento de imagem;
- Otimizar o programa C# de modo a evitar erros causados por funções criada pela VideoRay

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MELO, I. A.; FAVILLA, J. S. *Sistema de Controle e Monitoramento de um ROV*. Dissertação (de Graduacao) — Universidade de Brasilia, Faculdade de Tecnologia, Universidade de Brasilia, Brasilia, DF, 2015. 74p.
- [2] SOUZA, E. C. *Modelagem e controle de veiculos submarino nao tripulados*. Dissertação (Mestrado) — Escola Politecnica da Universidade de Sao Paulo, Sao Paulo, 2003. 168f.
- [3] FOSSEN, T. I. *Guidance and Control of Ocean Vehicles*. [S.l.]: John Wiley & Sons, 1994.
- [4] MÉTODO dos Mínimos Quadrados. Data de acesso: 20/06/2016. Disponível em: <<http://axpfep1.if.usp.br/otaviano/Apresentacao.pdf>>.
- [5] REN, T. I. Morfologia matemática.
- [6] DO, K. D.; PAN, J. *Control of Ships and Underwater Vehicles*. [S.l.]: Springer, 1994.
- [7] VIDEO Ray Pro: Operator Manual. Estados Unidos, 2010. 210p.

ANEXOS

I. DESCRIÇÃO DO CONTEÚDO DO CD

Descrever CD.

II. PROGRAMAS UTILIZADOS